Defense Technical Information Center Compilation Part Notice

ADP012337

TITLE: JACK: A System for Building Holonic Coalitions

DISTRIBUTION: Approved for public release, distribution unlimited
Availability: Hard copy only.

This paper is part of the following report:

TITLE: KSCO 2002: Second International Conference on Knowledge Systems for Coalition Operations

To order the complete compilation report, use: ADA402533

The component part is provided here to allow users access to individually authored sections of proceedings, annals, symposia, etc. However, the component should be considered within the context of the overall compilation report and not as a stand-alone technical report.

The following component part numbers comprise the compilation report:
ADP012330 thru ADP012354

UNCLASSIFIED

JACK: A System for Building Holonic Coalitions

Martyn Fletcher

Agent Oriented Software Ltd
Mill Lane, Cambridge CB2 1RX, UK
martyn.fletcher@agent-oriented.co.uk

Abstract. Coalitions between military and non-government organisations to manage operations other than war (OOTW), e.g. earthquake evacuations or food distribution to refugees require sophisticated knowledge management, decentralised control, and the ability to conduct flexible negotiations. Holonic systems are a research topic well suited to these requirements as they treat each organisation as an autonomous cooperative entity that simultaneously adopts a part-whole relationship within the coalition. This paper discusses a model of coalitions based on the application of holonic principles. The paper also outlines how this model could be implemented using JACK, the flagship software development product from Agent Oriented Systems.

1 Introduction

Arthur Koestler initially proposed the principles of holonics (Koestler, 1967). He postulated that many biological and social organizations display simultaneously part-whole relationships. In other words, every entity is self-contained, while concurrently being an individual member of a larger collective. Thus each entity (or *holon*) must act autonomously and cooperatively to achieve the goals of itself and the wider system. Hence the entire system can be seen as a *holarchy*, i.e. a recursive hierarchy or heterarchy of holons with no centralized control, which relies on collaboration among holons to achieve the system's goals. These generic ideas have been expanded on and delivered into agile manufacturing scenarios, and much has been learnt of holons' behaviour. Now it is time to apply these abstract concepts and the experiences gained in deploying such systems into other domains. A very suitable domain is coalition management as part of operations other than war (Tate, 1999). Here every military force and non-government agency can be viewed as a holon. The holarchy structure is then created in a 'bottom-up' manner via the aggregation of holons (royal air force, red cross and so on) to satisfy the requirements and services needed for handling the crisis.

A suitable foundation for implementing such holons is the agent-based development environment JACK from Agent Oriented Software (AOS, 2001). JACK is a realization of the belief-desire-intention model of agency and is one of the very few industry-strength systems for building autonomous-agent and team-based applications. This commercial product has a history of solid implementations through being deployed into defence, air traffic control and telecommunications environments. The utilization of JACK will provide a firm foundation for experimenting with agent-based coalition ideas during OOTW (Maughan, 2001) (Thomas, 2000). In the paper we illustrate how JACK can be applied to build, manage and control our new vision of holonic coalitions. The paper is structured to reflect these conceptual design and implementation issues, together with providing a simple illustrative example.

2 Conceptual Model of Holonic Coalitions

Holonic systems represent a novel paradigm for addressing some of the most critical problems encountered by military, charity and non-governmental organisations as they come to grips with the 21st century theatre of relief and humanitarian operations. These problems include:

- The demand from stricken governments and aid charities to have their specific relief/humanitarian requirements delivered to the crisis region with *short* 'request-to-deployment' times. People cannot wait a year for shelter, food or medical supplies to be delivered, or be evacuated from a hostile environment; they need it in 2 days.
- The need to support *mass customisation* of OOTW efforts, i.e. 'relief-to-order' rather than having dedicated military and non-government agencies on permanent standby ready to be deployed anywhere around the globe. This helps the agencies to regularly react to rush operations and new relief specifications.
- The need to have *tightly and loosely integrated* cooperation between agencies and hold/exchange appropriate private, protected and public knowledge.
- The requirement to cope with a hybrid combination of operational *variety* and *volume* within a single crisis area. Agencies are discovering that there is a need to distribute food to 1,000,000 refugees and conduct military actions against an enemy simultaneously. Traditional thinking and technology is not geared to this imbalance.

The benefits of applying holonic technology to OOTW include, but are not limited to:

- The holonic model helps the various agencies and military forces to make *maximal use* of available personnel, transport capacity, resources and assets to satisfy current/anticipated demand for relief. In other words, the system is able to support the re-allocation of tasks in a dynamic coalition through intelligent processes, reasoning, cooperation and negotiation – see (Shehory, 1998).
- Holonics treats alterations in coalition configurations, relief requirements, personnel, transport schedules and so forth as 'business as usual'. Moreover a holonic model reacts to the removal of, as well as introduction of new agencies, missions and information management facilities in a graceful fashion. In other words, the system is *agile* and does not crash due to changes in the operational environment.

Centralized solutions to controlling the coalitions between civil, government, charity and military organisations that satisfy such relief/humanitarian demands do not work since they are slow to react, impose operational bottlenecks and are a critical point of failure. Holonics is a decentralized 'bottom up' approach and provides principles to ensure a higher echelon of responsiveness and handling of system complexity. The building blocks (or components) of a holonic coalition architecture are called *holons* to reflect the fact that these entities behave simultaneously in an autonomous and cooperative fashion. Holonics is not just a new technology, but rather it is a system-wide philosophy for developing, configuring, and managing the next generation of OOTW where *flexibility* is paramount.

2.1 The Holonic Coalition System Architecture and Inter-Holon Cooperation

Coalitions unite people and organizations that share a common purpose. This section contains information on a few of the ideas that work toward awareness and improvement of holonic coalitions. The objective of a holonic coalition is to "attain in OOTW the benefits that a holonic system architecture has provided to intelligent manufacturing". Koestler observed the dichotomy of 'part-ness' and 'whole-ness' in natural systems (e.g. ant colonies), and devised the term ***holon*** from the Greek word ***holos*** (signifying whole) with suffix ***on*** (a particle, as in prot***on***). These generic principles have been studied in an intelligent manufacturing context to make production of high-variety low-volume artefacts more agile (Fletcher, 2001). Here we apply these same principles, and some of the experience gained as a result of these studies, to operations other than war. We model each charity (e.g. Red Crescent), civil government (e.g. local fire service) and military (e.g. Navy) agency as an autonomous cooperative holon. These agencies may be from different countries, represent diverse political/cultural/religious beliefs, have access to distinct resources/knowledge and may harbour resentment at being commanded by a military organisation. As discussed by (McFarlane and Gruver, 2001) within a manufacturing context, a holon is a basic building block in a holonic system. We propose that by applying these abstract ideas, there are the following holon types in a holonic coalition:

- *Agency holons* provide all the generic resources active in the OOTW system. Each of these agency holons is an entity (often a specialization of a particular class) that performs an action over an item. Such actions include those needed to transport, and disseminate relief materials to refugees and control the evacuation of people from hazardous environments. The items encompass food, trucks and refugees. Such agency holons include charities, military bodies, police forces, food collection companies, aircraft leasing companies, medical institutions etc.
- *Demand holons* represent the requirements of operations like relief work, peacekeeping and so forth. The requirements often originate from either external bodies (e.g. a stricken government), from other departments within an active organisation (for example the Army asking the Navy to supply a ship) or from anticipated need (for instance when the forecasts for a country indicate that a crop will fail next year then it is wise for an aid agency to stock pile food in readiness). These holons also provide knowledge on how to achieve the mission objectives, can offer expert advice, and may also act as an information server to disseminate knowledge among the other holons in the coalition. Each can be re-used in the scope of different operations and each could negotiate with various agency holons in order to secure the desired services. In other words, each demand holon is an active entity responsible for performing the crisis management work correctly and on time, while explicitly capturing all data and information processing needed for a specific job. Such demand holons might represent the need to evacuate 100,000 people after an earthquake and give multiple options how this could be achieved (e.g. by aircraft quickly, or alternatively via road slowly and so need a temporary shelter).

We hypothesise that the entire holonic coalition system can be modelled as a *holarchy*, namely a recursive aggregation of *cooperation domains* (see below). These cooperation domains solve a set of decomposed and inter-related OOTW tasks. Every task is modelled as a demand holon. The notion of a holarchy (see Figure 1) simplifies our architecture because we only need consider the structure of a single cooperation domain and the interactions agency and demand holons have through it. Using this holarchy principle, a simple *holonic team* is constructed to manage each cooperation domain. The members of this team could be either agency holons or other sub-teams (in a recursive manner). The lowest level holons are always *agency holons.* The structure created by this holarchy is specific for each crisis being managed and can be dynamic because:

1. Agency holons arrive/leave when their schedules or commitments change.
2. Demand holons enter/exit when their corresponding crisis task or knowledge is required or no longer needed.

Agency holons respond to task requests from the cooperation domains' demand holons that they are interested in participating within. Therefore either interaction is carried out (through the existing cooperation domain) or new crisis management tasks are generated (as demand holons) according to these responses. If a crisis management task cannot be executed due to a lack of an agency's resources (for instance inadequate equipment or peoples' skills) then the task may be altered. Otherwise a new functional component could be introduced into a holon to provide the necessary resource and so satisfy the cooperation domain's requirements.

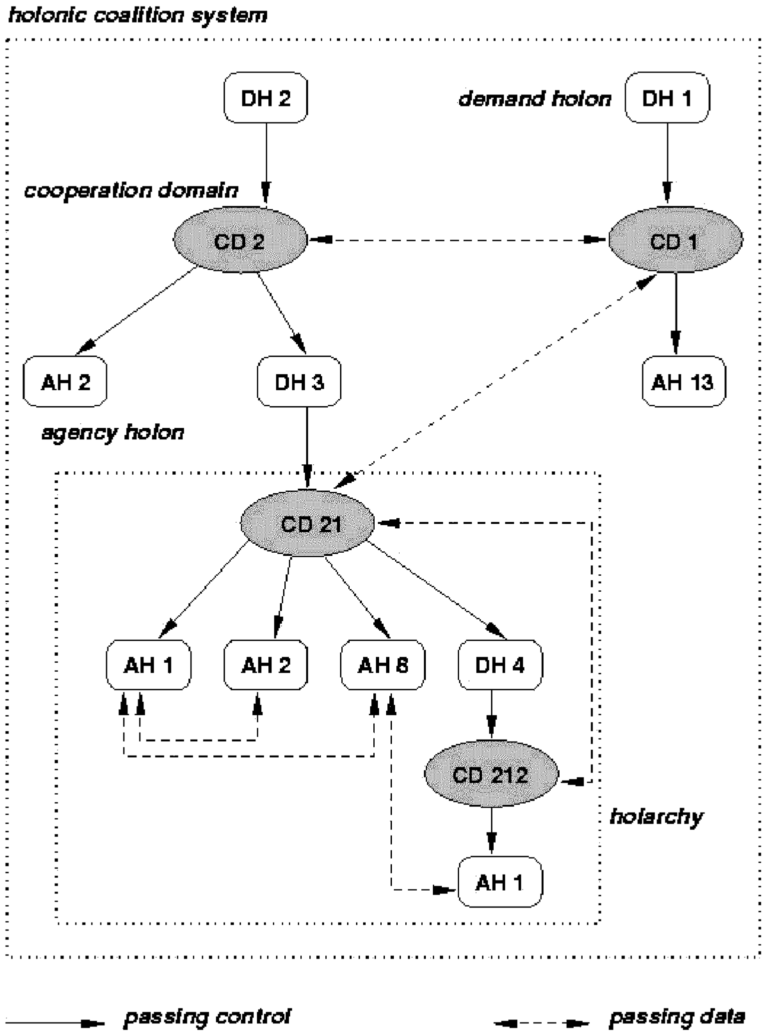

Figure 1: Coalitions, Cooperation Domains and Holons.

A cooperation domain is a logical space through which: (i) agent-based holons communicate and operate, and (ii) a context is provided where holons may locate, contact and interact with each other. We assert that a cooperation domain cannot exist by itself, and that all cooperation domains must be dynamically generated by the needs and services of individual holons. The following premises are valid with respect to cooperation domains:

- A holonic coalition system must contain at least one cooperation domain.
- An agency holon can be simultaneously a member of one or more cooperation domains.
- A cooperation domain can only exist if it has: (i) a demand holon; plus (ii) one or more member agency holons.

A cooperation domain comprises the following key elements:

- Coordination and information management facilities. These can be handled by a demand holon acting with the role of a coordinator to administrate a joint task, and retain/disseminate knowledge among agency holons.
- Data structures through which holons may write and read knowledge to control cooperation, e.g. querying the value of a variable that indicates the status of a joint food distribution task by the Red Cross and Air Force.
- Logical framework for connecting together heterogeneous holons. We model this property using a temporary alliance between a coordinator (demand) holon and one or more cohort (agency) holons that support:
  - Decision making mechanisms and rules to aid holons' task planning, scheduling, negotiation, information dissemination and so forth.
  - Facilities to monitor the status of distributed tasks, and take appropriate corrections to compensate for any anomalies during execution of actions within this task.
- Physical communication platform. We assume holons pass messages using a reliable transport mechanism.

Holons can join a cooperation domain, query attributes associated with a domain, exchange information amongst one another through the cooperation domain, and depart the domain when their crisis management tasks are completed. Furthermore we visualize that a cooperation domain supports a 4-phase protocol (agreement, planning, interaction and termination) to provide a formal model of inter-holon collaboration for joint actions.

2.2 The Intra-Holon Architecture

As stated earlier, we define an agency holon as an autonomous system having a compulsory knowledge-based element and an optional physical element. For instance the Red Cross has a people to negotiate and decide how to best deploy its resources (knowledge-based element), while its resources include medical personnel, food, trucks and so on (physical element). A demand holon has no physical element. Moreover suitable interfaces to humans, other holons and the OOTW environment must also be present. In terms of its behaviour, an agency holon's knowledge-based element consists of an intelligent control system (ICS) and a processing system interface.

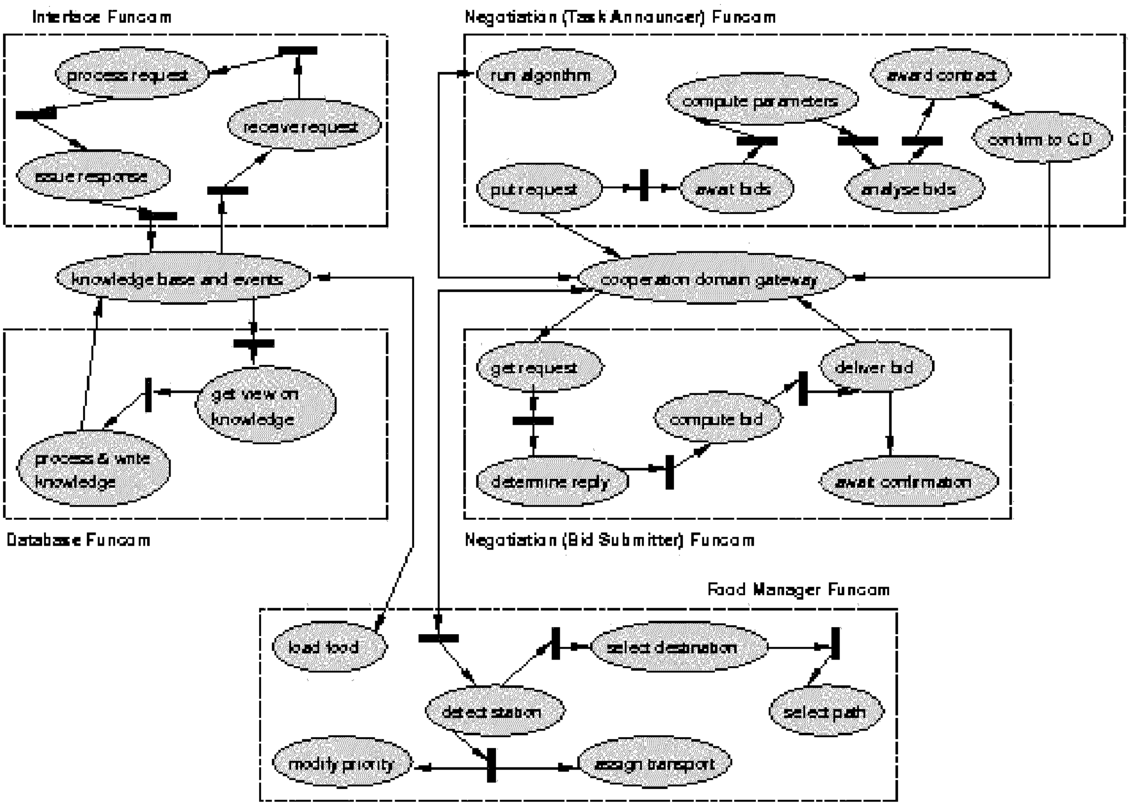

Figure 2: Generic and Application-Specific Funcoms in Demand/Agency Holons.

The ICS is responsible for the holon's internal functionality through a set of procedural rules and decision making functions. The ICS also supports cooperation via inter-holon interfaces, acquaintance modelling and so forth. In short, the intelligent control system of an agency holon is modelled as an *agent* as understood in multi-agent systems (Pechoucek, *et al.*, 2001). The processing system interface provides the individualistic skills of the agency holon and

is responsible for the relief, humanitarian and military functionality according to rules and operating strategies imposed by the ICS. The processing system interface is divided into a collection of *functional components* (or *funcoms*) necessary to realize a wide variety of skills needed in operations other than war. Each funcom has independent control over its activities. For example, an Army agency holon (as part of a food distribution task) contains the subsequent functional components:

- *Load-Food*: Loads/unloads volumes of food (e.g. bags of rice) from the local sea docks or airfield into trucks.
- *Detect-Station*: Identifies the status (i.e. in the range full to starved) of food distribution stations.
- *Select-Destination*: Chooses which food distribution station should get the next delivery of goods.
- *Select-Path*: Chooses the best possible route from food entry points to the selected destination station.
- *Modify-Priority*: Alters the foods' priority or the requirements of peoples' need for this particular food type.
- *Assign-Transport*: Assigns the task of transporting the food to a given truck.

Agency holons' funcoms are designed so that they contain all the knowledge and skills required to manage operations effectively and efficiently. In this sense, we regard knowledge as being the database tuples, trigger rules events, and the beliefs, desires and intentions of the associated agents. The justification for requiring this knowledge is that it supplies both a structured semantic representation to generalize a quantity of items related to the holonic coalition system, and an anchor point to which a future implementation can be attached. Such knowledge may be classified as being local (i.e. obtained from monitoring the state of the adjacent environment), regional (i.e. that which is received from neighbouring holons) or global (namely data acquired from a directory holon). In this sense skills are the operations needed by an agency holon to utilize and maintain such knowledge, together with the corresponding manipulation of their resources (e.g. food) as they are received, transported, stored and disseminated. These skills are modelled as application-specific funcoms. As described in the Figure 2, there are also some general-purpose functional components that are used to build up each holon. These generic funcoms ensure that the agency/demand holon has sufficient autonomy and cooperation (the negotiation funcom) and can form suitable association with other agency/demand holons (the interface funcom):

- The **Negotiation (Task Announcer)** functional component operates within the demand holon to implement the first half of the entire negotiation cycle. Within the scope of the aforementioned holarchy, this funcom negotiates with the funcoms in agency holons to agree, plan, execute and terminate an operation to satisfy the OOTW demand. To achieve this planning etc, the task announcer funcom supports a number of protocols like the contract net protocol (CNP), various styles of auction, or a market economy; we consider the CNP here. The task announcer funcom is the element of the demand holon that starts a negotiation cycle; that means: (i) putting into the cooperation domain a request for some OOTW task to be accomplished, (ii) computing its parameters using the proprietary algorithms, (iii) waiting for the bids submission, (iv) analysing the bids from the various military/charity/non-government organisations, (v) running its proprietary algorithm to evaluate these bids and award the contract, and (vi) putting the confirmation of the contract into the cooperation domain.

- The **Negotiation (Bid Submitter)** functional component operates within the agency holon to implement the other half of the negotiation cycle. Briefly, this funcom replies to the task announcement to complete a negotiation cycle, in particular this means: (i) getting the OOTW task request from the cooperation domain, (ii) accepting it and deciding if reply to it using its proprietary algorithms, (iii) computing the bid using its private knowledge base and its proprietary algorithm, (iv) delivering the bid, and (v) waiting for the confirmation that award the contract to the agency holon.

- The **Interface** functional component operates within every agency and demand holon and allows the interaction between: agency-to-agency holons, agency-to-demand holons and demand-to-demand holons. The most complex and complicated of these interactions is the agency-to-agency exchange because some charity and military organisations do not want to share all their private knowledge within every other agency within the holarchy. To acquire essential information from another organisation, the agency holon uses an "information protocol" that offers a mechanism to call for information that is proprietary to the other agency holon. Of course this request can be rejected or false information can be given depending on how the two agencies consider each other. As proposed in some of the holonic manufacturing system literature (Van Brussel, *et al.*, 1998), a centralised *'staff'* holon can be used to suggest a solution (for example the allocation of how much food each of three charities should distribute in the relief operation over the next week) and the agency holons ask for such compromises and information using their respective interface functional components.

Using the above definitions, we can now model the coalitions between military and non-government holonic agents. To clarify this point, the next section presents an illustrative example of how holonic coalitions in OOTW can work.

3 An Illustrative Example

Once a basic infrastructure among the relevant agencies is established, new forms of holonic coalitions and advanced cooperation between these agencies will naturally emerge. The satisfaction of the demand holon's functional requirements in the OOTW theatre will also begin to be comprehensively supported. In particular, holonic coalition formation requires mechanisms to facilitate the controlled 'introduction' of a military, charity or non-government body (e.g. the US Marines to act as a food distributor) into the 'territory' of the relief operation (e.g. a humanitarian effort in a West African country) without impinging on the roles and attributes of its partners (e.g. the Red Crescent for giving food to Muslims, the local police force to guard food supplies and Christian Aid disseminate food to non-Muslim refugees). An initial illustrative example of this introduction is sketched out in Figure 3. The introduction is properly supported by the above holonic model, and administrates the access to selected (authorised by the agreements made when joining the relevant cooperation domain) subsets of the necessary resources (for instance food stocks, distribution personnel, trucks, helicopters etc). But this process may assume more extensive forms. Consider the case where the US Marine commander wants to 'open a window' on coalition partners to get an overall picture of how well the food distribution process is going and even have an interference on, i.e. supervise from distance and in cooperation with native-speaking local police, the dispatching processes.

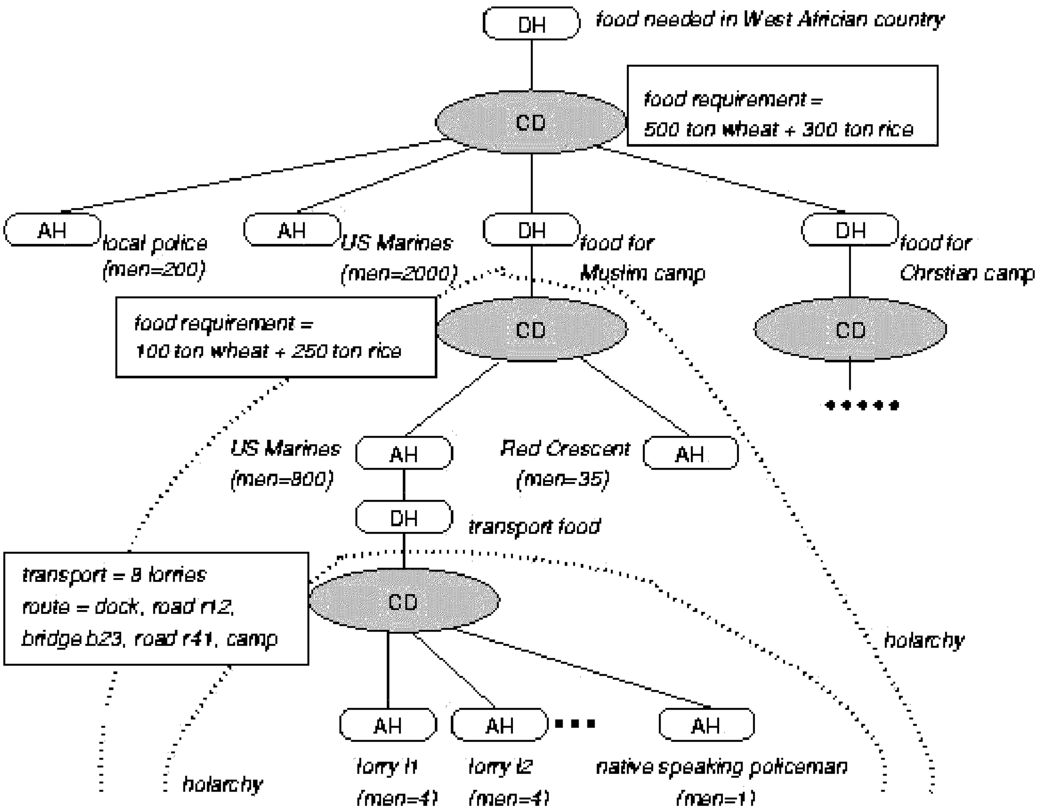

Figure 3: An Example of Holonic Coalitions – Initial View.

Such supervision represents a collection of inter-agency holon and demand-to-agency holon activities including:

- The dispatch and execution of task requests to move food from point A (e.g. the docks) to point B (a camp set up by the Red Crescent for Muslim refugees fleeing from an on-going guerrilla war in their home town).
- The monitoring of execution, for instance knowing accurately how much wheat and rice has been moved to each refugee camp, what are camps' expected demands and what additional resources will become available.
- Error diagnosis and recovery, for instance discovering that a bridge along the main route to a refugee camp has been destroyed and deciding to instead transport 2/3 of the food via a different route and use helicopters to transfer the remaining 1/3.

When viewed in a decentralised operations-other-than-war environment, the concept of *coalitions* emerges. If the coalition demands the collaboration among multiple agencies, each acting as an independent body and part of a team, each being located in remote places, being managed in different ways and adopting distinct internal structures and rules, then we have *holonic coalitions.* Therefore we need a model like that presented in section 2 with agency and demand holons, funcoms and cooperation domains to support such holonic coalitions. The design of a proper

support system for holonic coalitions can benefit from contributions coming from a number of topics. These areas are at present addressed by research communities with little interaction between them. The four key contributing areas to holonic coalitions are holonic manufacturing systems, multi-agent systems, political science and defence/military studies. This paper focuses on the application of the generic principles and experiences coming from holonic manufacturing systems and how these concepts could be implemented using multi-agent technology – see next section. We leave the investigation of the roles political science (to model charity and non-government bodies) and defence/military studies (to represent military organisations) play in holonic coalitions for later work.

Coalitions between charities and military organisations have been addressed for a few years, mainly for small-scale OOTW exercises where the military body is in overall command. However the growing number of peace keeping, relief and humanitarian operations around the globe, and the requirement for military bodies not to impose on the charities and non-government agencies has opened up new opportunities for coalitions due to their lower cost, even distribution of workload and widespread acceptability. What makes holonic principles most appealing as a basis for coalitions is how they model each operation's demands and every agency involved as autonomous cooperative entities that can operate independently, collaborate and exchange knowledge in a structured fashion to achieve the OOTW objectives. However the application of holonic principles to coalitions suffers from several problems:

1. Is a solution based on today's implementations of agents, cooperation domains, funcoms, or an amalgamation versatile enough to solve the multitude of diverse problems encountered when building real holonic coalitions? The response must be a decisive no; at present, it is not. These models support some aspects of holonic behaviour very well (e.g. the concept of encapsulation of software executing at the real-time level of control), and some issues slightly less well (for instance the lack of ability to dynamically decompose a demand for some relief work into atomic tasks without pre-defined static rules). But let us not fool ourselves by saying that everything is finished. For example if we use a combined solution then where is the boundary to be set between one agency holon's autonomous activities and the actions it must manage via a loosely-coupled coalition among bodies that have distinct goals. Furthermore how are these independent technologies (with no obvious shared protocols) to interact in a cohesive manner?
2. When reasonably practical and complex OOTW domains are considered, high levels of heterogeneity are expected in the available agencies and the demands put upon them. This interoperability requirement, together with the volume, accuracy and type of knowledge to be exchanged among agencies, can degrade the agility, robustness and saleability of demand/agency holons operating in the holonic coalition system.
3. Coalitions are characterised by the short and irregular durations, also they are negotiation intensive due to the peer-to-peer level of collaboration between agencies where the military body cannot impose on the charities. These attributes mean that the coalitions can often suffer from low levels of trust, limited private resources forthcoming from non-government bodies and restricted exchange of knowledge between coalition 'partners'. This raises new challenges in what concerns the reliability and efficiency of the implemented holonic coalition system and its dependence upon the characteristics of the constituent allies.
4. The composition of the environments where holonic coalitions are to be executed are potentially unstructured and unknown. This means that it is inadequate to resort to deterministically programmed systems or monolithic centralised systems. Complementary, the increased use of military bodies to support peace keeping, refugee evacuation and so forth requires multiple interaction periods, of varying durations, with agencies that: (i) might not behave in a altruistic fashion; or (ii) have their own goals to achieve beyond the present coalition's scope.

In order to cope with the mentioned difficulties, an approach based on autonomous agent and multi-agent system technologies has been developed. Multi-agent systems originate from research into Distributed Artificial Intelligence (DAI) (Hewitt, 1981) and use mentalist approaches to problem solving by imitating human actions and interactions. These concepts are often based on speech acts (Searle, 1969) or the belief-desire-intention (BDI) model. Like people, such models are inherently unpredictable, can be unstable and may make wildly different decisions based on uncertain knowledge. Hence agents may not be best suited for every real-world coalition case, especially whose where there exist safety critical and secrecy constraints of tasks. Yet their benefits are numerous (e.g. fault-tolerance, dynamic reconfiguration etc) and so their exploitation is ensured. The BDI model was initially introduced as the foundation for single-agent architectures by (Bratman, *et al*., 1988) and was developed further by, amongst others, (Rao and Georgeff, 1995). Since its conception, the BDI scheme has become a solid foundation for research into multi-agent architectures and their application to several problem domains. The scheme defines both:

- An agent's internal processing through a set of mental categories with a control framework for the rational selection of action plans to satisfy goals using some knowledge of the environment.
- A team (as part of *Team Oriented Programming*) that encapsulates multiple agents into a group with a concerted goal and set of beliefs. This group then has a specific coordinated activity to perform, and so assigns roles to independent agents to get the joint task achieved.

These principles have been further extended by Agent Oriented Software (AOS, 2001), made into a commercial product called JACK (Howden, *et al.*, 2001) and has been successfully applied to control various application domains including a manufacturing cell at the University of Cambridge's Institute for Manufacturing (Jarvis, *et al.*, 2001). From which we have gained a lot of valuable experience in deploying agent-based holons. Here we use some of this experience to build coalitions based on holonic ideas using JACK.

4 Building Holonic Coalitions with JACK

JACK Intelligent Agents is an agent-oriented development environment that is built on top of, and is fully integrated with, the Java programming language. JACK consists of:

- *JACK Agent Language (JAL).* JAL encompasses Java and is used by software engineers to build holonic coalition systems by providing a 'super-set' of agent-oriented constructs. JAL extends Java by: (i) Providing new base classes, methods and interfaces; (ii) Extending Java syntax to support new classes, declarations and reasoning method statements; and (iii) Providing semantic extensions to support agent-oriented execution.
- *JACK Agent Compiler.* This compiler pre-processes JAL source files and converts them into standard Java. This can then be compiled into Java Virtual Machine code and executed upon some target holonic system.
- *JACK Agent Kernel.* This kernel provides all the runtime facilities to execute these holonic agent constructs (written in JAL).

The structure of a JACK agent and how it works is as follows: Each agency and demand holon is an instance of a particular agent class, and interacts with its physical OOTW environment through a set of functions that read data in from people in the physical operations theatre (e.g. information is supplied by charity people with Internet mobile phones and PDAs, or through military personnel with laptop computers and secure satellite communications systems etc) and write instructions out to the same human beings. Every agent representing an agency holon has one or more capabilities modelled as application-specific funcoms (for example fault diagnosis, scheduling and the food manager as shown in Figure 2) that it can perform. Each capability encapsulates a number of goals (or desires), plans (or intentions), knowledge (or beliefs) and event templates that the agent will react to.

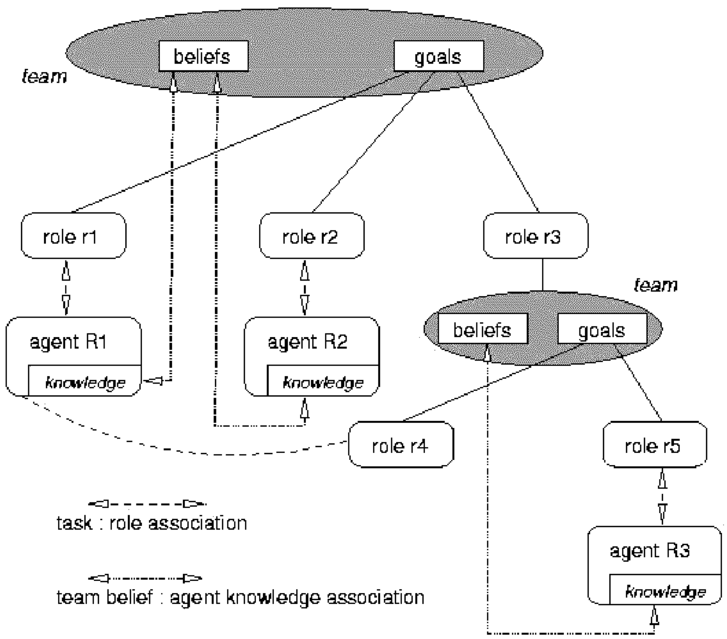

Figure 4: Team Oriented Programming.

When this agent-based agency holon is instantiated into the holonic coalition system, it will wait until it receives an event that it must respond to, or is presented with a goal. Agent-based demand holons have equivalent functionality for handling where and when the operations are needed and their parameters. Events are used to support reactive behaviour in the agents while goals are utilized to focus an agent's proactive behaviour. When it receives such an event (or goal) then it searches for and then executes a suitable plan(s) to handle an instance of this event type. Such event handling may be either synchronous or asynchronous to when it was posted. The execution of this plan may demand: (i) the exchange of data and instructions with other holonic agents via a suitable protocol, (ii) interaction with the agent's private permanent database relations, or (iii) manipulation of other Java-based non-permanent data structures. The plan being executed can create other sub-tasks, which in turn may generate sub-sub-tasks, and so on,

thus creating a recursive hierarchy, that is adequate for modelling the conceptual holarchy organisation outlined above. Each plan may either succeed or fail, in which case the agent may attempt to execute another plan.

A *simple team* is an extension of JACK and allows for the definition of agent groups where coordination of joint activities is distributed across team members. In our holonic framework, each cooperation domain is modelled as a team in order to help with group functionality and share workload. JACK also supports teamwork by providing a set of concurrency management and event handling functions. This team-engineering concept is flexible and does not impose rigid criteria on the formation of multi-agent collectives or on the dissemination of beliefs among team members. These ideas are shown in Figure 4. The system developer has the freedom to choose the subsequent team attributes to build holonic coalitions:

What the team is capable of doing, i.e. what is the team's overall goal? In our holonic coalition context, the goal is to ensure optimal and efficient movement of food from docks to various refugee camps through the alliance of distinct non-government and military bodies as they enter, leave and reconfigure their actions/interactions in the coalition.

What are the *roles* of individual member agency and demand holonic agents within the scope of achieving this team's goal? Here the roles assigned can be either:

- One demand holon per multi-body operation to represent the task's parameters, and some functionality to monitor and advise on how the task should be handled by the available agency holons. In our earlier example, the demand holons include 'food needed in a West African country' at the highest level, going down the recursive tree, to 'transport food to Muslim refugee camp'.
- One coordinator agency holonic agent responsible for managing the operation and one or more coordinatee agency holonic agents that obey the commands given them by the coordinator. This is a master slave relationship where the coordinator identifies the potential operation, isolates what options can be taken, assigns tasks, issues commands to other agency holons to achieve the goal and monitors these actions to ensure success.
- Multiple negotiator agency holonic agents responsible for collaborating together to discover and execute the best overall food dissemination strategy. This is a peer-to-peer relationship where all the agency holonic agents have an equal vote on what joint action to take.

What is the assignment of roles to actual team members? Here we allocate the coordinator role to agency holon *US Marines* and coordinatee roles to *Local Police, Red Crescent* and *Christian Aid*. The allocation of the roles is also bound to any resource-dependent constraints on the task. For **hard** resource-bound tasks, e.g. some chilled food must be delivered by time *t1* using refrigerated lorry *l12*, the action must always be completed by the specified due time. While for **soft** resource-bound tasks, the actions must be completed to a certain percentage of occasions by the set finish time and using the requested resource. To reflect this distinction, the role is given to a particular agency holonic agent the completion time, resource specifications and the ratio is also presented. For **hard** resource-bound tasks the ratio is 100%, for **non** resource-bound tasks (i.e. common agent-based actions) the ratio is 0%, and for **soft** resource-bound tasks the ratio is in the range 1% …. 99%.

What functional components are needed to form this particular class of team? Here we can say that the coordinator agency holon must have suitable plans to discover, determine, asses potential food logistics and the ability to formulate a solution. While the coordinatee agency holons need to execute the distribution plan assigned to them and report their status. In other words, a number of algorithms are needed at each different type of holonic agent.

When is a team willing to take on a particular role within the confines of another team? Namely what is the recursive nature of these agency holon aggregations. Let us illustrate by example, suppose military organisation *US Marines* enters into a cooperation domain with non-government charity *Christian Aid* and the resolution of this food transport operation is that *Christian Aid* should stop moving medical equipment while *US Marines* uses some common lorries to proceed through their shared hostile working envelope, e.g. an area with an on-going armed conflict. This means that *Christian Aid* will not meet its expected food delivery schedule at its refugee camp to distribute food the people. Hence *Christian Aid* must resolve this secondary coordination problem (e.g. getting the food delivered to the camp via another transport option like using police vehicles via a subordinate team) without impinging on the solution of the top-level team.

How is behaviour across the team members to be coordinated? What techniques and methods are needed to ensure synchronised mutually-agreed actions are taken throughout the team community. Here we hypothesise that a simple publisher subscriber model will suffice: the agency holon representing the *US Marines* writes knowledge to the cooperation domain, suitable monitoring functions are invoked, the allied bodies like *Christian Aid* are informed, and act according to their prescribed role. More sophisticated contract bidding, auction or economic market solutions could be used especially when the agency holonic agents might wish not to disclose private knowledge.

How is the knowledge in the team to be encoded, disseminated, and replicated between autonomous team member agents? We postulate that suitable ontologies, multi-casting, and consistency protocols can be called upon respectively. No system stands alone, and so a holonic coalition system built in JACK must work both on its own and integrate tightly with other agent-based solutions. JACK agents are not compliant with the existing standards from the Foundation for Intelligent Physical Agents (FIPA, 2001). Therefore you cannot put together intelligent software agents constructed in JACK and other systems like CPlanT (Pechoucek, *et al.*, 2001) in a haphazard way and expect them to work. There are some alternatives that can be used to ensure these heterogeneous agents can integrate smoothly:

- Send and receive events through a common operational environment, e.g. the battlefield and OOTW theatre that both types of agents can observe.
- Send/receive knowledge to a shared database that generates appropriate events that both agent types can utilise.
- Have a bridge agent to convert messages from FIPA-compliant format to a suitable format for JACK agents to use, and vice versa.

A team is defined in terms of the roles played by its members and so may be composed of either autonomous agents (agency/demand holonic agents) or subordinate teams (with the lowest level of this recursive organisation always being an agency holonic agent). In short, the teams principle allows for the encapsulation and engineering of coordinated activity among heterogeneous holonic agents. The teams concept extends the notion of autonomous agents into multi-agent systems via the association of tasks with roles. Yet each agency holonic agent remains autonomous and is privately responsible with determining how its plans can best satisfy the role(s) assigned to it. We now present some JAL code for implementing such holonic coalitions. These coalitions are relatively well defined, with several roles, and involve a reasonable amount of parallelism.

```
package aos.simpleteam.core;
import aos.simpleteams.rt.*;

team HolonicCoalition extends SimpleTeam {
        #requires role CoordinatorHolon coord_h;
        #requires role FoodTransporters[2] trans_h;
        #requires role FoodDistributer dist_h;
        #requires role Interrupter int_h;

        #uses plan Transport_and_Distribute_Food;
}
```

We note that the team has two distinct food transporters; otherwise the roles are singular. Since only the food transporters are distinct (i.e. ground-based transport – lorries, and air-based transport – helicopters), other roles may be filled by the same team member or by different team members according to the formulation of the plan. For the FoodDistributer role, we model two alternatives. If the actual OOTW holonic coalition is less complex then the distribution role can be handled by a simpler team that directly performs the task. For complex coalitions requiring very large movements of food, a larger distribution team maybe needed as modelled by ComplexDist below.

```
team SimpleDist extends SimpleTeam {
        #performs role FoodDistributer dist_h;

        #uses plan GetDistributer;
}

team ComplexDist extends SimpleTeam {
        #performs role FoodDistributer dist_h;

        #requires role PoliceLiaison pl;
        #requires role TechLead lead;
        #requires role FoodDispatcher disp;
        #requires role Administrator admin;

        #uses plan AcquireDistributer;
}
```

The larger food distribution team thus includes an explicit role separation for police liaison, technical leadership, dispatch of food, and administration. We continue the illustration of JACK's team-based features by suggesting a team plan for the HolonicCoalition team according to the following principles:

```
team_plan Transport_and_Distribute_Food extends TeamPlan {
        #uses team HolonicCoalition team;

        body () {
                @team_achieve(team.coord_h.ManageCoalition());
                @parallel() {
                        @team_achieve(team.trans_h[1].Unload-Food());
                        @team_achieve(team.trans_h[2].Detect-Station());
                }
                @parallel() {
                        @team_achieve(team.int_h.Contact-Local-Leader());
                        @team_achieve(team.dist_h.AcquireDistributer());
                }
        }
}
```

The reader may verify that the team plan represents an equal assignment model, with necessary jobs within the food transportation process broken down into parallel tasks. For instance, the group of people attached to the US Marines truck unit (distributor holon 1) unloads the food [Unload-Food] from the dock while concurrently the helicopter unit (distributor holon 2) selects which refugee camp this food consignment should be sent to [Detect-Station]. We note that the plan includes a declaration that enables access to the team structure. The team plan is a sequence and parallel set of actions to be performed by the team entity (in other words by the holonic coalition) with the goal of coordinating how and when these actions are to be performed by the team members. From this example, though it is far from complete, we can highlight some features of JACK's team oriented modelling approach and also point out some of its shortcomings:

- It allows for the description of team-based and autonomous agent-based activities in a clear and concise fashion.
- It enables the abstraction of what needs to be done from how it is to be accomplished, and facilitates for the team plan to be constructed without considering how the roles are to be fulfilled. This can be clearly observed by having two very different groups of holonic agents that can perform the FoodDistributer role.
- It shows how rapidly even simple team-oriented programming can become complex. Building a robust team application (in our case for holonic coalitions in OOTW) demands good software engineering practices, knowledge and computer-based tools.

We hypothesise that designing the same holonic coalition example without JACK's team-oriented programming concepts – namely developing the plans and messages for conventional autonomous agents – would easily result in a system that is very complicated and almost impossible to maintain. A change to the team's behaviour (i.e. a modification to the demand holon's requirements in a specific cooperation domain) would then impact many agency holons, and the centralised specification would be lacking. At the same time, although the above example illustrates a neat team structure within a holonic coalition, together with a realistic knowledge and activity flows, it implements an idealised view that may be difficult to realise in pragmatic military/non-government operations. For instance, as coalition management may run in parallel with the other activities, there may also be intricate control structures spanning the carious controlled activities (e.g. regular status reporting and so forth). It is not immediately clear whether the team modelling approach sufficiently enables such coordination to be captured in a natural manner. We now make some concluding remarks.

5 Concluding Remarks

There is a 21st Century demand for agile combinations of non-government and military agencies to conduct disaster relief work, peace-keeping and provide humanitarian support to people in stricken areas. These areas are often the scenes of on-going fighting and so operations other than war must be conducted in a way that protects civilian workers while not impeding battle objectives. This is a difficult balance to achieve. Owing to these requirements, application of flexible coalitions will typify how many organisations involved in war avoidance operations will have to operate. The paper has hypothesised that agent-oriented holonic behaviour could realise this next generation of decentralised knowledge-based coalition systems. We envisage that the introduction of "holonic" ideas into such OOTW coalitions will lead to a significant increase in the following characteristics:

- Robustness and stability in the face of disturbances: the system of coalesced agencies has monitoring methods to replace holons and reschedule their tasks. Also message passing is supported by resilient platforms.
- Adaptability and flexibility to rapid change: interactions between governments (as part of treaties like NATO) control the behaviour of holons for given tasks by specifying cooperation strategies as and when needed. Strategies use high-level commands and encourage holon transparency and accountability. To be scaleable, holons interact through logical spaces called cooperation domains. Holons can create, join, leave and destroy cooperation domains at run-time to satisfy the individual requirements of the crisis.
- Efficient use of available resources: holons manage their own failures and take appropriate actions to compensate for any lost effectiveness. Holons may also balance load amongst themselves to ease any strain.

(Koestler, 1967) brilliantly envisaged what such holonic behaviour should look like with holon/human societies, distributed intelligence and system construction based on every social entity being simultaneously a whole system and part of a larger structure. Translation of these abstract ideas into the technical realm of OOTW demands much research: all the problems must be identified, solved and software implemented. Only then can we provide the complete holonic solution ready for such agencies to take onboard. This paper addressed some initial ideas on how a model of holonic coalitions could be constructed. We also demonstrated how this model could be implemented using JACK, one of very few commercial-strength implementations of the BDI autonomous agent model. Though some of the concepts and coding presented here are speculative, their importance should not be overlooked. Such ideas are needed as part of a more comprehensive methodology for building and deploying coalitions.

References

- "Agent Oriented Software Ltd", (2001), http://www.agent-oriented.co.uk/.
- Bratman, M.E., Israel, D.J., and Pollack, M.E., (1988), "Plans and Resource Bounded Practical Reasoning", *Computational Intelligence*, vol 4, no. 4.
- Van Brussel, H., Valckenaers, P., Bongaerts, L. and Peters, P. (1998), "PROSA: A Reference Architecture for Holonic Manufacturing Systems", *Computers in Industry*, vol. 37, pp. 155-274.
- "FIPA – Foundation for Intelligent Physical Agents", (2001) http://www.fipa.org.
- Fletcher, M., Norrie, D.H. and Christensen, J.H. (2001), "A Foundation for Real-time Holonic Control Systems", *Journal of Applied Systems Studies*, special issue on Holonic and Multi-Agent Systems, vol. 2, no.1, Cambridge International Science Publishing.
- Hewitt, C. (1981), "Open Information Systems Semantics for DAI", *AI journal*, vol 4, no. 2.
- Howden, N., Ronnquist, R., Hodgson, A., and Lucas, A., (2001), "JACK Intelligent Agents - Summary of an Agent Infrastructure", Proceedings of the 5th ACM International Conference on Autonomous Agents, Montreal, Canada.
- Jarvis, D., Jarvis, J., Lucas, A., Ronnquist, R., and McFarlane, D.C., (2001), "Implementing a Multi Agent Systems Approach to Collaborative Autonomous Manufacturing Operations", Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Tuscon, USA.
- Koestler, A. (1967), "*The Ghost in the Machine*", Arkana, London.
- Maughan, D. (2001), "Overview of Dynamic Coalitions Programme", DARPA Information Technology Office, http://www.darpa.mil/ito/research/dc/index.html.
- McFarlane, D.C., and Gruver, W., (2001) "Workshop on Distributed and Intelligent Manufacturing", Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Tuscon, USA.
- Pechoucek, M., Barta, J. and Marik, V. (2001), "Acquaintance Model Based Coalition Planning in Humanitarian Relief Operations – Progress Report", Technical report of the Gerstner Laboratory, Czech Technical University in Prague.
- Rao, A.S., and Georgeff, M.P., (1995), "BDI Agents: From Theory to Practice", Proceedings of the 1st International Conference on Multi Agent Systems, San Francisco, USA,
- Searle, J.R. (1969), "*Speech Acts*", published by Cambridge University Press.
- Shehory, O. and Kraus, S. (1998), "Methods for Task Allocation via Agent Coalition Formation", *Artificial Intelligence*, vol 101 (1-2), pp 165-200.
- Tate, A. (1999), "Proceedings of the International Workshop on Knowledge-based Planning for Coalition Forces", Edinburgh, UK.
- Thomas, J.P. (2000), "The Military Challenges of Transatlantic Coalitions", Adelphi paper 333, IISS publications, http://www.iiss.org/pub/prap333.asp.